ÍNDICE

FASE 5

DATATHON

MACHINE LEARNING ENGINEERING

DATATHON

Olá, estudante! Chegamos ao Datathon da Fase 5!

Nesta fase não teremos um Tech Challenge, mas sim um Datathon para **desenvolver todas as habilidades aprendidas durante o curso!** Nele, esperamos que você utilize as melhores técnicas aprendidas ao longo de nossa jornada para solucionar um problema real de uma das maiores redes de televisão brasileira, a Globo.

A Globo é uma empresa de tecnologia de mídia que produz e distribui uma vasta gama de conteúdos através de suas plataformas, os quais incluem vídeos, podcasts, portais de notícias, entretenimento e esportes. Com a enorme quantidade de itens gerados diariamente, torna-se inviável realizar a curadoria manual desses conteúdos, além de ser um desafio garantir sua distribuição eficiente para milhões de usuários que acessam essas plataformas.

Nesse contexto, os sistemas de recomendação se tornam peças-chave para personalizar a experiência do usuário, promovendo conteúdos que se alinham aos seus interesses.

Lembrando que sistemas de recomendação são projetados para sugerir ao usuário itens com base em seus hábitos anteriores de consumo

As recomendações podem ser feitas com algoritmos simples, como a exibição dos conteúdos mais populares em determinado período, ou com técnicas mais avançadas, como os algoritmos baseados em conteúdo (content-based), que consideram as características do próprio item para gerar sugestões. Outra abordagem bastante comum é prever o próximo item que o usuário irá consumir com base em seu histórico.

Um desafio comum a todas essas abordagens é o problema do **cold-start**, que ocorre quando um novo usuário ou um novo item não possui informações suficientes para alimentar o sistema de recomendação. Esse problema é agravado no caso de notícias, devido à rápida atualização desse tipo de conteúdo.

Por exemplo: em plataformas como o Globoplay, é possível recomendar conteúdos antigos como a série A Grande Família, que continua sendo popular. No entanto, o mesmo não se aplica ao G1: recomendar uma notícia de dois anos atrás seria inapropriado, visto que a recência é um fator essencial no consumo de notícias.

OBJETIVOS

Seu desafio é desenvolver esse modelo de sistema de recomendação e realizar o deploy dele utilizando as técnicas aprendidas no curso. Nesse cenário, surge o desafio de fornecer recomendações personalizadas para cada usuário com base nos dados de notícias do G1, predizendo qual será a próxima notícia que ele vai ler. Alguns pontos importantes devem ser considerados:

- formações devem ser tratados de forma diferenciada em rela-
  ção a perfis mais completos?

- O conceito de recência é essencial para garantir que as reco-
  mendações de notícias sejam relevantes e oportunas.

Seu projeto deve conter os seguintes passos. Não se esqueça de gravar um vídeo para explicar todo o projeto e a disponibilização do repositório do GitHub.

1. Treinamento do modelo.
2. Salvamento do modelo.
3. Criação de uma API para previsões.
4. Empacotamento com Docker.
5. Testes e validação da API.
6. Deploy em ambiente produtivo: API local ou nuvem (a nuvem é opcional).

Caso tenha acesso a algum serviço de nuvem (AWS, GCP), faça o deploy do container em um serviço como AWS ECS ou Google Cloud Run.

Link da base de dados e dicionário de dados

https://drive.google.com/file/d/13rvnyK5PJADJQgYe-VbdXb7PpLPj7lPr/view

SOBRE OS DADOS

O conjunto de dados para este desafio foi dividido em treino e validação.

Conjunto de Treino

O conjunto de treino está disponibilizado em diferentes pastas, cada uma contendo informação complementar. Os arquivos treino_parte_X.csv, em que X é um valor de 1 até 6, consistem das colunas:

1. **userId**: id do usuário.
2. **userType**: usuário logado ou anônimo.
3. **HistorySize**: quantidade de notícias lidas pelo usuário.

5. **TimestampHistory**: momento em que o usuário visitou a página.
6. **timeOnPageHistory**: quantidade de ms em que o usuário ficou na página.
7. **numberOfClicksHistory**: quantidade de clicks na matéria.
8. **scrollPercentageHistory**: quanto o usuário visualizou da matéria.
9. **pageVisitsCountHistory**: quantidade de vezes que o usuário visitou a matéria.

Além desses arquivos, a pasta de treino contém uma subpasta denominada de itens. Ela contém a seguinte informação:

1. **Page**: id da matéria. Esse é o mesmo id que aparece na coluna history de antes.
2. **Url**: url da matéria.
3. **Issued**: data em que a matéria foi criada.
4. **Modified**: última data em que a matéria foi modificada.
5. **Title**: título da matéria.
6. **Body**: corpo da matéria.
7. **Caption**: subtítulo da matéria.

Este conjunto de treino consiste em dados de usuários reais da Globoplay. Eles forem coletados até uma data limite T (a maior data em todo o conjunto TimestampHistory).

O seu objetivo é gerar um ranking para a coluna history. Ou seja: quando um usuário loga, é necessário prever quais serão os próximos acessos dele. Observe que o mesmo userId está tanto na validação quanto no treino.

Boa sorte, pessoal! Contem conosco caso haja alguma dúvida no desenvolvimento do projeto. 🍀

CONTE-NOS SOBRE A SUA EXPERIÊNCIA

O que você achou do conteúdo deste capítulo?

QUER SE APROFUNDAR MAIS?

VEJA AS REFERÊNCIAS!

ÍNDICE